Using Function Approximation in Personal Point-of-Interest Recommendation

Bilian Chen[1], Shenbao Yu[1], Jing Tang[2], Mengda He[2], and Yifeng Zeng[2]

[1] Department of Automation, Xiamen University, China
blchen@xmu.edu.cn
[2] School of Computing, Teesside University, UK
{J.Tang, M.He, y.zeng}@tees.ac.uk

Abstract. Point-of-interest (POI) recommender system encourages users to share their locations and social experience through check-ins in online location-based social networks . A most recent algorithm for POI recommendation takes into account both relevance and location diversity. The relevance measures users' personal preference while the diversity considers location categories. There exists a dilemma of weighting these two factors in the recommendation. The location diversity is weighted more when a user is new to a city and expects to explore the city in a new visit. In this paper, we propose a method to automatically adjust the weights according to user's personal preference. We focus on investigating a function between location category numbers and a weight value for each user, where the *Chebyshev* polynomial approximation method using binary values is applied. We further improve the approximation by exploring similar behavior of users within a location category. We conduct experiments on five real-world datasets, and show that the new approach can make a good balance of weighting the two factors therefore providing better recommendation.
Key words: POI recommender system · Location category · Tuning parameter

1 Introduction

Social networks connecting various types of users have played important roles in our daily life where people share their experience in a real-time manner. Meanwhile, by exploiting information that grows over time in social networks, many service providers have developed useful service portals, like friend-matching, recommendation and advertisement, that return much benefit to users. Recently, with the rapid development of positioning techniques, such as Global Position System (GPS), Wireless Fidelity (WI-FI), etc., intelligent terminal equipments are commonly used and users can report their physical locations in social networks. Social networks with additional records of users' geographical positions, namely location-based social networks (LBSNs), attract more and more users as well as service providers such as Foursquare, Gowalla, Facebook, etc..

In LBSNs, users can report positions of point-of-interests (POIs) in their travel, and publish their experience and tips for the POIs. All the users' information, called *check-ins* in LBSNs, have been largely used for a POI recommendation purpose. The recommendation mechanism lies in the rationale: users who have similar travel experience in the past may share common POIs in new visits. Hence a user would be advised to visit the POIs where his/her friends have travelled. This has inspired a line of research on POI recommendation techniques and their improvement such as user-based collaborative filtering [18] and matrix factorization method [3,5]. Later on, Yuan *et al.* [17] improved the traditional recommendation method by adding temporal influence. Since most of the methods mainly focus on the recommendation through users' similarity - their relevance on the travel experience, they tend to generate a set of homogeneous POIs in the recommendation set.

To improve the diversity of POIs in the recommended set, Chen *et al.* [4] introduced a new factor, namely information coverage, that measures the diversity of POI's categories in the candidate set. Consequently, one POI needs to be scored through a tradeoff between two factors: relevance and information coverage. This is always hard in a general optimization problem. There exists a dilemma between two factors, i.e., increasing one will result in decreasing the other, and vice versa. For example, the relevance factor should be less weighted when it is the first time for a user to visit a city while it should be stressed if the user has visited this city for many times and expects to explore some specific locations in a new visit.

In this paper, we aim to automatically weight the two factors in a POI recommendation thereby achieving personal recommendation. We observe that users' desire to visit new POIs' categories is reduced over time. This exactly satisfies the law of diminishing marginal utility. Hence we formulate the weighting of two factors as one function approximation problem in which the function is to represent personal travel preference on the exploration of POIs' categories in a new visit. We take a principled method by using *Chebyshev* polynomials to approximate the function. Given sufficient information of users' visits to POIs, it can be proved in a theoretical way that the approximation converges to the real function of personal preferences. However, the existing check-in data often can't provide a full profile of individual user's travel preference. This compromises the quality of function approximation using Chebyshev polynomials, which results in reduced recommendation accuracy.

We take a further step to alleviate the issue of data sparsity in order to improve the function approximation. In the parameter estimation of Chebyshev polynomials, we use data of a set of similar users instead of a single user in the process. We cluster the similar users through their previous travel experience that implies their personal preferences on the POI exploration. However, as spotted in the experimental study, the clustering is very sensitive to the user-POI distribution in check-in data. We analyse the performance of the proposed techniques for personal POI recommendation on five real-world datasets.

The rest of this paper is organized as follows. In Section 3, we formulate our problem. The POI recommendation is described in Section 2. In Section 4, we propose two approaches based on Chebyshev polynomials to solve our problem. Section 5 demonstrates the performance of our method upon five large scale real-world datasets crawled from *Foursquare* and *Gowalla*. We review related works in Section 6 and conclude this paper in Section 7.

2 Background: POI Recommendation

We present top-K location category based POIs (LC-POIs) recommendation framework suggested by Chen *et al.* [4]. The recommendation considers two factors to score potential POIs: one is POI relevance and the other is information coverage.

2.1 POI Relevance

The relevance computation follows user-based collaborative filtering methods [1]. Let $l \in L$ be a POI in the POI set L. For a given user u, the relevance score for a POI is computed in Eq.(1).

$$R(l) = \alpha \times c^t_{u,l} + (1 - \alpha) \times c^s_{u,l} \tag{1}$$

where $c^t_{u,l}$ considers similar experience between users and can be calculated by user-based collaborative filtering methods, $c^s_{u,l}$ counts the spatial influence of POIs and α is the tuning parameter.

The score for a set of POIs is a sum of the relevance score for all elements as shown in Eq.(2).

$$R(L) = \sum_{l \in L} R(l) \tag{2}$$

2.2 Information Coverage

Information coverage improves the recommendation quality by considering diversity of location categories. In general, the larger degree of covering all categories, the more information the set of POIs can provide in the recommendation.

Let $A = \{a_1, \cdots, a_q\}$ be a set of location categories. The information coverage score of a set of POIs is computed in Eq.(3).

$$I(L) = \sum_{a_q \in A} \omega_{a_q} cov_{a_q}(L) \tag{3}$$

where ω_{a_q} is the weight of category a_q and is calculated through TF-IDF(Term Frequency-Inverse Document Frequency) technique, and $cov_{aq}(L)$ is the measurement of the degree to which a set of POIs covers a_q. Details about information coverage computation shall be found in [4].

2.3 Top-K LC-POIs Recommendation

Given the two factors defined above, the top-K LC-POIs recommendation problem for a given user is to find a set of POIs that maximize the scoring function $\sigma(L)$ by computing their relevance and information coverage in check-in data. Specifically, we need to solve the following multi-objective optimization problem:

$$\textbf{Given} : D, K, \beta, u$$
$$\textbf{Object} : max_{L \subseteq D, |L|=K} \quad \sigma(L) = (1-\beta) \times R(L) + \beta \times I(L)$$

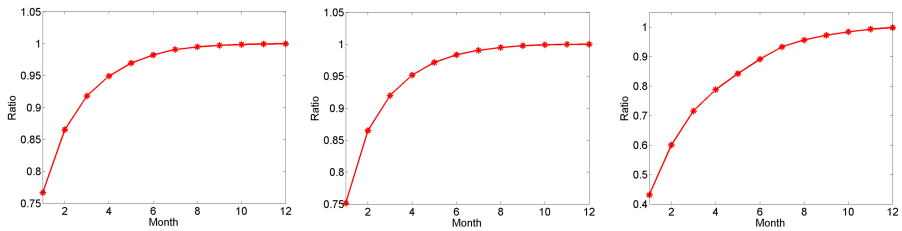
where $L \subseteq D$ traverses all POIs and $\beta \in [0,1]$ is the adjustable parameter, which makes a balance between the POI relevance and information coverage.

The top-K LC-POIs problem is proved to be NP-hard, and the solution can be found using the greedy algorithm with pruning strategies as shown in [4].

3 New Problem Formulation

To find a good way to balance the relevance and information coverage factors in POI recommendation, we first analyze the check-in data in *Foursquare* made within Singapore and in *Gowalla* made within Austin and Stockholm. As demonstrated in [4], users would like to explore new types of locations, e.g., restaurants, bar and mall, that have not been visited before. When a user has covered most parts of the city, his desire to explore new location categories decreases. Fig. 1 confirms this observation in the real-world datasets containing check-ins over the time period. The slope of each curve gradually decreases over time. For a given month, we compute the ratio of location categories that a user has visited previously to all ones visited during the considered time period, and then report the average value for all users. As time goes, the degree of users' desire for knowing this city is reduced. This phenomenon can be explained by the law of diminishing marginal utility in economics [14], which says that the marginal utility of each homogenous unit decreases as the supply of units increases (and vice versa).

The parameter β reflects the tradeoff between POI relevance and information coverage. We employ the well-defined measurements to evaluate the scores of POI relevance and information coverage, namely recommendation accuracy and diversity of the recommended POIs, respectively. If an improper β is selected, it may lead to an extreme case, e.g., the highest diversity with the lowest accuracy. In general, β should not be set at a high level when a user has various visiting records in one city. Obviously, the diversity is closely related to the number of location categories. Hence, we connect the parameter β to the category number denoted by cn. In contrast that Chen *et al.* set the value of β as a uniform constant [4], we aim to adjust β automatically according to the number of POIs category that the user has already visited. Consequently, our task is to find a function, saying $\beta = f(cn)$, to establish the relationship between β and a location category number for each user.



(a) Austin (b) Stockholm (c) Singapore

Fig. 1: The trend for users to explore location categories over time

4 Chebyshev Polynomial Approximation

In this section, we adopt the *Chebyshev* polynomial approximation method to fit the function $\beta = f(cn)$. To proceed the function approximation, we analyze check-in data and estimate the parameter value for each data instance.

4.1 Analyzing Check-in Data

In LBSNs, the check-in data D maintains the visiting records of each user in a time period. A sample is provided in Table 1. For a given user u, we extract the following information for further use:

- A set of check-in days in ascending order, denoted as T [3];
- Given a certain day $d \in T$, the category number cn_d is counted during time period $[0, d]$, which serves as an input value for Chebyshev polynomial approximation; the set of category numbers $\{cn_d\}$ denoted as N;
- A set of POIs that user u will visit in time period $(d, \max(T)]$, denoted as POI^d_{true}. Subsequently, the total POI number K_d and the diversity div^d_{true} of POI^d_{true} are obtained.

Table 1: Sample of User Check-in Sequences

User-ID	POI-ID	(Lati.,Long.)	Day-ID	Category
0	22847	(30.23,-97.79)	625	Indian Restaurant
1	420135	(30.26,-97.74)	569	Office
2	18417	(30.24,-97.75)	566	Coffee Shop
...	...	...	...	...

[3] The elements in T are not continuous.

4.2 Parameter Estimation

We give an estimated value for parameter β corresponding to the diversity of the set POI_{true}. The top-K LC-POIs recommendation method introduced in Section 2.3 is used in the estimation approach (line 4). Increasing weight of parameter β will lead to a wide range of location categories in a POI recommendation. An estimated value β_{est} could be determined if the POI recommendation meets the user's most current flavor (lines 5-6). We describe the parameter estimation in Algorithm 1.

Algorithm 1 Estimating Parameter β

Input: D, cn, u, d, POI^d_{true}, $threshold$
Output: β^d_{est}

```
1: Initialize β^d_est = 0;
2: Compute : K_d, div^d_true
3: for β = 0 → 1 do
4:     Compute : div^d_est corresponding to Top-K LC-POIs recommendation(K^d,
       β);
5:     if | div^d_est - div^d_true| < threshold then
6:         β^d_est ← β;
7:         break;
8:     end if
9: end for
10: return β^d_est;
```

4.3 Function Approximation

Chebyshev polynomials form a special class of polynomials especially suited for approximating other functions. Any function $f(x)$ may be approximated by a weighted sum of these polynomial functions.

$$f(x) = \frac{a_0}{2} + \sum_{i=1}^{\infty} a_i \, T_i(x)$$

where $T_i(x) = \cos(i \cos^{-1}(x)), i = 0, 1, \cdots$ are Chebyshev polynomials and $a_i = \frac{2}{\pi} \int_{-1}^{1} \frac{f(x)\, T_i(x)}{\sqrt{1-x^2}} dx$ are the coefficients. In practice, we could truncate the infinite series and get an approximation of function $f(x)$ [9].

Basically, a function approximation requires some certain values of x and $f(x)$. In our case, given a category number cn, its corresponding true value β is, however, unknown. We first compute an estimated value for β in Section 4.2. Then we toss a biased coin with the weight β_{est} to determine whether or not to accept the value. Hence, we may employ the application of Chebyshev polynomials using a binary value to approximate our function.

For a given user u, let $\hat{f}(cn)$ be the approximation function based on the set of samples $S = \{(cn_d, y_{cn_d})\}$, $\forall d \in T$, where $y_{cn_d} \in \{True, False\}$. The total number of samples is $|N|$. For an implementation purpose, we scale the values cn_d $(\forall d \in T)$ to the range $[-1, 1]$.

Algorithm 2 Chebyshev Polynomial Approximation

Input: D, T, u, $P = \{(cn_d, \beta^d_{est})\}$, $\forall d \in T$
Output: Coefficients C_i and function $\hat{f}(cn)$

Initialize $C_i = 0$, $\forall i = 0, 1, \ldots, m$;
for all $d \in T$ **do**
  for $i = 0 \to m$ **do**
    if y_{cn_d} = True **then**
      $C_i \leftarrow C_i + \frac{T_i(cn)}{\sqrt{1-(cn)^2}}$;
    end if
    if y_{cn_d} = False **then**
      $C_i \leftarrow C_i - \frac{T_i(cn)}{\sqrt{1-(cn)^2}}$;
    end if
  end for
end for
Set: $\hat{f}(cn) = \psi(N) \times (\frac{C_0}{2} + \sum_{i=1}^{m} C_i \times T_i(cn))$;
Set: $\hat{f}(cn) \leftarrow \hat{f}(cn) + 0.5$;
return $\hat{f}(cn)$;

Suppose m is the number of Chebyshev polynomials and $T_i(cn), i = 0, 1, \ldots, m$ are the Chebyshev polynomials. Let $\psi(N)$ be a function of $|N|$. For a given user u, Algorithm 2 presents the details of computing coefficients C_i and approximated function $\hat{f}(cn)$. We further formulate the algorithm property.

Theorem 1. *Given complete data, Algorithm 2 returns the targeted function $\hat{f}(cn)$.*

Proof. We approximate the function based on the set of samples $S = \{(cn_d, y_{cn_d})\}$, $\forall d \in T$, where cn_d refers to the category number that a user has been visited at a certain time, and $y_{cn_d} \in \{True, False\}$ is the decision variable to accept or reject the estimated value for parameter β. Let $\hat{f}(cn)$ be the estimated function by Algorithm 2, and $f(cn)$ be the real one, which can be computed by a combination of Chebyshev polynomials:

$$f(cn) = \frac{a_0}{2} + \sum_{i=1}^{\infty} a_i \, T_i(cn)$$

Analogous to the proof in [15], we can show that

$$\begin{aligned} C_0 &= a_0 - 1, \\ C_i &= a_i, \forall\, i = 1, 2, \cdots . \end{aligned}$$

Hence,

$$\hat{f}(cn) = 0.5 + \frac{C_0}{2} + \sum_{i=1}^{\infty} C_i \, T_i(cn)$$

which completes the proof.

4.4 Improved Function Approximation By Grouping Users

As users in the check-in system have very limited histories, data sparsity becomes an issue in approximating the function. The number of data points used for Chebyshev polynomial approximation in Algorithm 2 is very small. It would be helpful if we can explore additional valuable information to support the approximation. This thought is also mentioned by Saha *et al* [15]. They carried out an experiment focusing on the effect of the number of decision samples available on the accuracy of the model developed and showed that increasing the number of samples improved the approximation quality.

For a given user u, we search for his similar users and obtain more data to help the user u get a fairly close function. The similarity is based on the degree of users' desire for knowing this city, which is the closeness between the increment of location category numbers that two users have visited previously at each time slot[4]. To be specific, we obtain similar users of the user u through the following steps:

- Calculate the increment of location category numbers that user u has visited previously at each time slot, and collect them in a vector $\Delta cn_u = (\Delta cn_{u,1}, \Delta cn_{u,2}, \cdots, \Delta cn_{u,k})$, where k is the total number of time slots;
- Let U be the set of all users. Compute the similarity matrix $M \in R^{|U| \times |U|}$, where each element of M is the cosine similarity between users u and v calculated as follows:

$$Sim(u,v) = \frac{\Delta cn_u \cdot \Delta cn_v}{\|\Delta cn_u\| \|\Delta cn_v\|} = \frac{\sum_{i=1}^{k} \Delta cn_{u,i} \cdot \Delta cn_{v,i}}{\sqrt{\sum_{i=1}^{k} \Delta cn_{u,i}^2} \sqrt{\sum_{i=1}^{k} \Delta cn_{v,i}{}^2}} \tag{4}$$

- Classify users based on the matrix M by the spectral clustering method [2], and then get the set of similar users of u, denoted as S_u.

Algorithm 3 presents the procedure for grouping users.

5 Empirical Study

To evaluate the performance of our algorithms for tuning parameter in the POI recommendation, we conduct a series of experiments on multiple real-world data sets. All the codes are implemented in JAVA, and all the numerical computations are conducted on a Windows PC with a 4-core Intel i5-4590 3.3GHz CPU and 8GB memory.

[4] The time slot could be a day, a week, a month, etc.

Algorithm 3 Chebyshev Polynomial Approximation by Grouping Users

Input: D, T, u, S_u, $P = \{(cn_d, \beta^d_{est})\}$, $\forall d \in T$
Output: Coefficients C_i and function $\hat{f}(cn)$

```
for all v ∈ S_u do
    for all d_v ∈ T_v do
        if d_v ∉ T then
            T ← T ⋃ {d_v};
            P ← P ⋃ {(cn_{d_v}, β^{d_v}_{est})};
        end if
    end for
end for
Execute Algorithm 2;
```

5.1 Experimental Setup

Datasets. Five real datasets are used in our experiments. One was collected from *Foursquare* which was made in Singapore between Aug. 2010 and Jul. 2011 [17], the others were collected from *Gowalla* which were made among four cities that including Austin, Stockholm, San Francisco and Dallas, the span of time is from Feb. 2009 to Oct. 2010 [6]. Each check-in has the aforementioned attributes listed in Table 1. For these datasets, we remove users who have checked in fewer than 5 POIs, and then remove POIs that are checked by fewer than 5 users. We summarize the characteristics of the check-in data in Table 2. For each user, we randomly mark off 20% of his or her visited POIs as testing data to illustrate the importance of tuning parameter. The remaining 80% as training data to obtain the function of category number $f(cn)$ via Chebyshev polynomial approximation whether the data is grouped or not, respectively.

Table 2: Statistics of the check-in data

City	No. of Check-ins	No. of Users	No. of POIs
Austin	183,795	4,515	4,729
Singapore	188,772	2,311	5,350
Stockholm	168,758	6,138	5,727
San Francisco	134,481	3,725	5,029
Dallas	6,366	2,103	2,931

Parameter Settings. Unless stated otherwise, number of Chebyshev polynomials m is set to be 5, and the threshold of Algorithm 1 is set to be 0.1. Category number $cn \in N$ is transformed in the range $[-1, 1]$ by the linear mapping $\hat{cn} = \frac{cn - mean(N)}{max(N) - mean(N)}$.

Metrics. To evaluate the recommendation method presented in Section 2.3 with adjustable parameter, we use three metrics, namely, precision@K, recall@K and diversity@K(denoted by pre@K, rec@K and div@K respectively), where K is the number of recommendation results, see, e.g., [4,17]. The pre@K measures how many POIs in the top-K recommended POIs correspond to the hold-off POIs in the testing data, the rec@K measures how many POIs in the hold-off POIs in the testing set are returned as top-K recommended POIs, and the div@K measures the category diversity of the recommended POIs based on the Shannon entropy [8]. In our experiment, we test the performance of $K = 5, 10, 20$.

We also employ the mean absolute error MAE as a measurement to show the accuracy of the category numbers we predict. Suppose that we have already had the approximation function $\beta_u = f_u(cn)$ for each user. For a targeted user u on a given day $d \in T$, let $r_{u,d}$ be the true value of category number in POIs that user u will visit during time period $(d, max(T)]$, the predicted category number $\hat{r}_{u,d}$ is calculated by the following steps:

- Obtain category number cn_d of POIs visited by user u during $[0, d]$, and the POI number K_d that user u will visit later.
- Compute $\beta_d = \hat{f}_u(cn_d)$;
- Get the set of POIs Q_d by top-K LC-POIs recommendation method with input values β_d and K_d;
- Count $\hat{r}_{u,d}$ corresponds to the set Q_d.

The $MAE(d)$ for time slot d is calculated as follows:

$$MAE(d) = \frac{\sum_{u \in U} |\hat{r}_{u,d} - r_{u,d}|}{|U_d|}$$

where U_d is a set of users who have check-ins In the day d. The overall MAE IS calculated by averaging the $MAE(d)$ values over all time slots:

$$MAE = \frac{\sum_{d \in T} MAE(d)}{|T|} \tag{5}$$

5.2 Experimental Results

We demonstrate the efficiency of our method for approximating function $\beta = f(cn)$ from two aspects. One is to measure the effect of adjustable parameter via pre@K, rec@K and div@K, while the other one is to measure the accuracy of predicted category numbers via MAE.

Effect of The Adjustable Parameter. As mentioned before, finding a way to balance between the accuracy and diversity is important. $\beta = 0$ means that the POI relevance is considered only, while $\beta = 1$ means that the information coverage is maximized in the top-K LC-POIs recommendation. Fig. 2 shows the effect of adjustable parameter with $K = 5, 10, 20$ on precision, recall and diversity, respectively. Note that the values of precision and recall in Fig. 2 are smaller than that of showed in [4] possibly because we remove the time-aware information. Compared to those that are obtained when $\beta = 0$ or $\beta = 1$, we observe that by adapting the parameter on each given day for every user, we can still maintain a high level of diversity in the recommendation results, while enjoying a good quality of precision and recall, which illustrates that the approximating technique in Algorithms 2 and 3 (denoted by CPA and CPAG, respectively) is reliable. It can be also seen from Fig. 2 that the overall performance of Algorithm 3 is much better since we have more data points to get a closer Chebyshev polynomial approximation, especially for Austin, Stockholm, San Francisco and Dallas. However, the performance of Singapore obtained by Algorithm 3 is abnormal, the reason will be discussed later in Section 5.2.

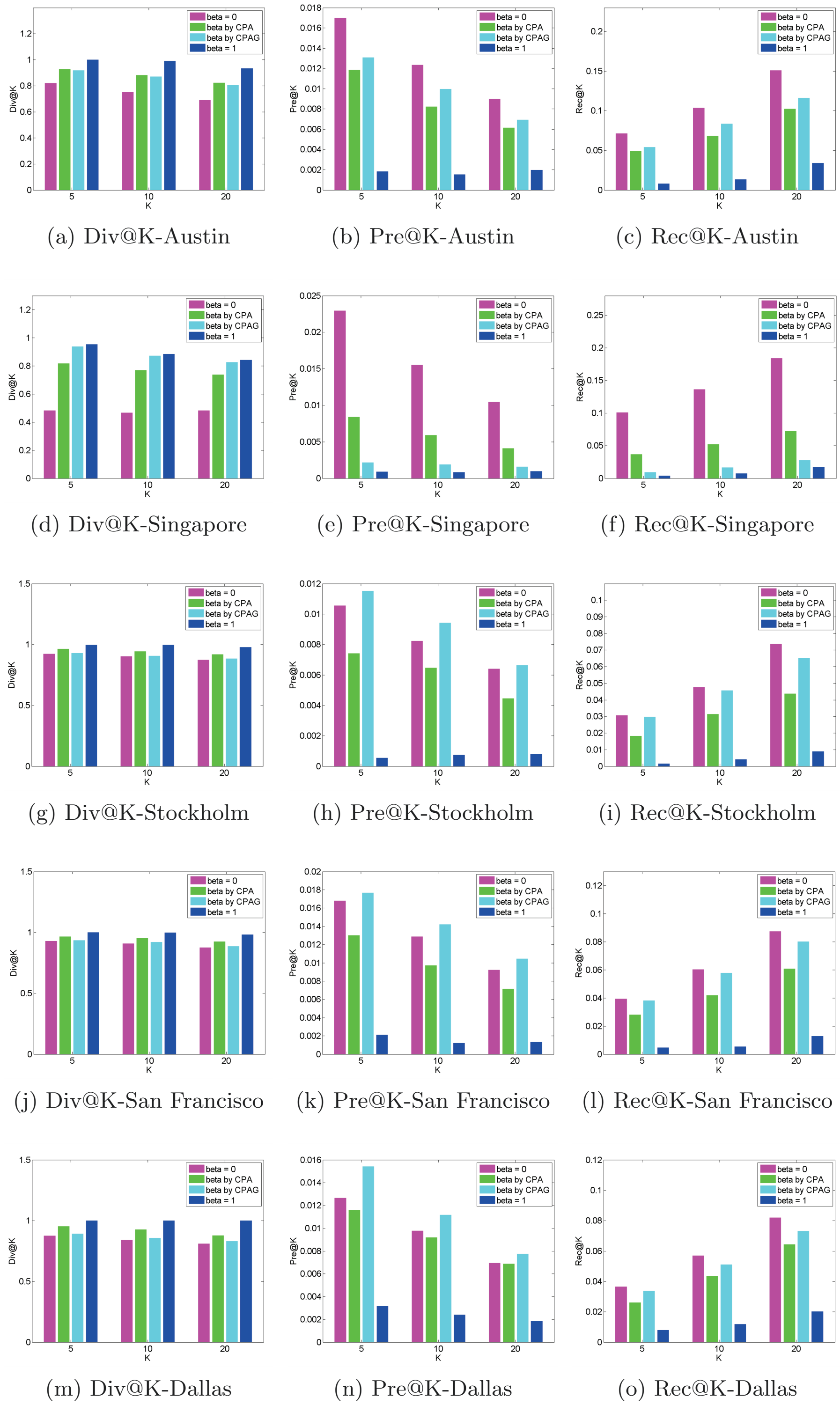
Accuracy of Predicted Category Numbers. We first present the prediction results of Algorithm 2. Fig. 3 shows the overall performance of forecasting category numbers for all users, illustrating the good forecasting quality of category numbers via our approach. For a given day d, the true value in Fig. 3 equals to $\sum_{u\in U} r_{u,d}/|U|$ and the predicted value equals to $\sum_{u\in U} \hat{r}_{u,d}/|U|$. We observe that the proposed technique can accurately predict the category number of POIs that users will visit. Furthermore, we calculate the metric MAE for each city in Table 3, indicating that there is only a small gap between predicted values and true values on average.

Table 3: The MAE values for five cities

City	Austin	Singapore	Stockholm	San Francisco	Dallas
MAE	1.41	2.23	0.84	2.01	1.86

In addition, we evaluate the performance of forecasting category numbers for every user. Fig. 4 shows three users' performance in Austin, indicating that the value β suggested by Algorithm 2 can track users' preference at any time, while it is not the case if β is randomly generated. It is particularly worth mentioning that Fig. 3 and Fig. 4 show that the category number of POIs that users will visit at a certain time Day decreases over time. That also confirms that the fluctuation of the ratio of category numbers follows the law of diminishing marginal utility explained in Section 3. We conclude that the results provided above suggest that tuning the parameter is necessary and Algorithm 2 generate reliable and promising performance.

Improved Accuracy of Predicted Category Numbers By Grouping Users In this part, we evaluate the prediction capability of Algorithm 3. In the implementation, unless stated otherwise, the number of groups for spectral



(a) Div@K-Austin (b) Pre@K-Austin (c) Rec@K-Austin

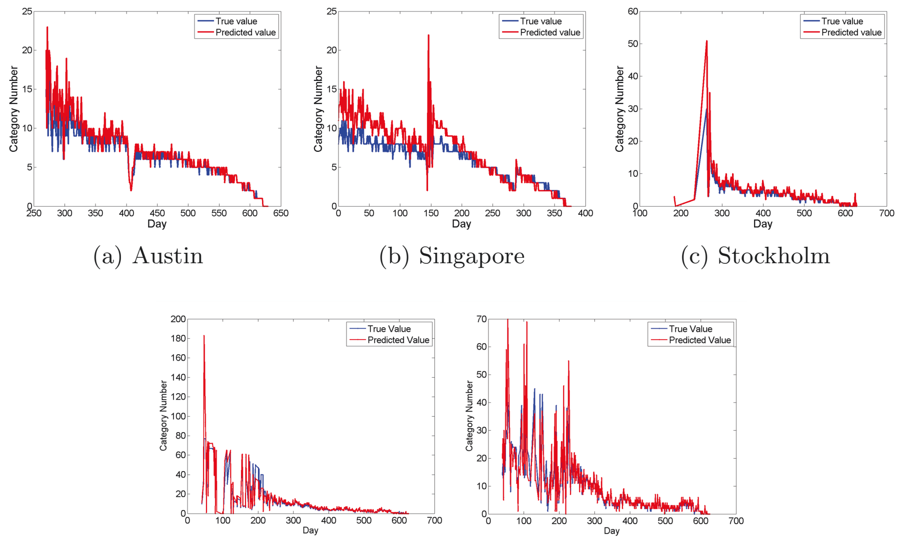
(d) Div@K-Singapore (e) Pre@K-Singapore (f) Rec@K-Singapore

(g) Div@K-Stockholm (h) Pre@K-Stockholm (i) Rec@K-Stockholm

(j) Div@K-San Francisco (k) Pre@K-San Francisco (l) Rec@K-San Francisco

(m) Div@K-Dallas (n) Pre@K-Dallas (o) Rec@K-Dallas

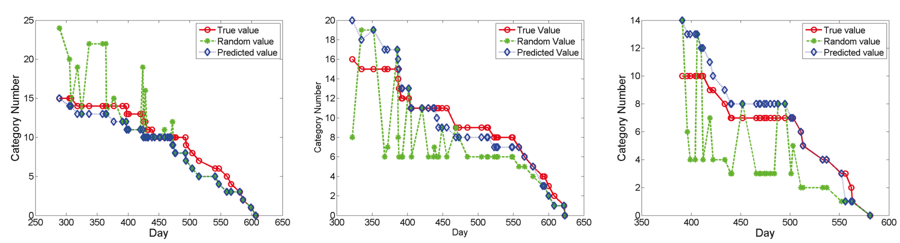
Fig. 2: Effect of the adjustable parameter



(a) Austin (b) Singapore (c) Stockholm

(d) San Francisco (e) Dallas

Fig. 3: Overall performance of forecasting category numbers



(a) User1-Austin (b) User2-Austin (c) User3-Austin

Fig. 4: Performance of forecasting category numbers for three users in Austin

clustering method is set to be 50 and the time slot is set to be one week. By grouping similar users in terms of behavior of increment of location category numbers, the average number of samples used for the approximation in Algorithm 3 does increase a lot, see Table 4 for details. Thus, the approximation of Algorithm 3 should be more accurate than that of Algorithm 2 in general. Figure 5 shows the comparison results of MAE values via the two approaches. The accuracy of predicted category numbers gets improved in three cities, that is, Algorithm 3 decreases the MAE value of Algorithm 2 by 39%, 48% and 28% in Stockholm, San Francisco and Dallas respectively. However, for the results of Austin and Singapore, they are both increased by 8% and 4%, respectively. Next, we explore the reasons for this deterioration.

Table 4: The average number of samples for approximation

City	Austin	Singapore	Stockholm	San Francisco	Dallas
CPA	8	7	14	7	7
CPAG	33	30	38	34	46

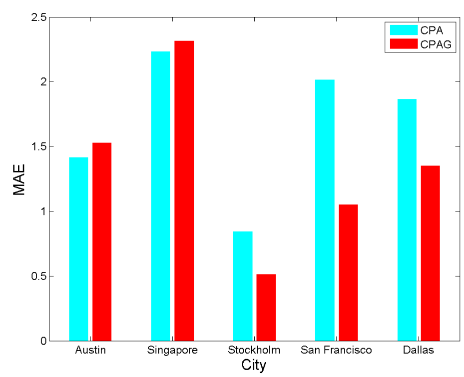


Fig. 5: Comparison results of the MAE values

We first look into the dataset of Austin after grouping users. There exists a large group whose number of users reaches 1152, accounting for about 25.5% of the total number of users. Even though we increase the number of groups when implementing the spectral clustering method, saying 100, 200 and 300 respectively, the maximum number of users in a single group always accounts for about 20% of the total one, see Table 5. The similarity of the behaviors of users in Austin is significantly large. Thus, in these special groups, every user owns more than 260 days of activity records on average in a year after grouping, it is too often to be in line with people's daily habits. Table 5 also shows that the MAE value for each case obtained by Algorithm 3 is increased by about 3-5%, illustrating that the technique of grouping users is not suitable for Austin.

Table 5: Results of the largest group

No. of Groups	No. of Users	No. of Check-in Days		Increased Ratio
		CPA	CPAG	
50	1152	8	306	5.2%
100	974	6	285	4.9%
200	922	5	264	5.3%
300	902	5	262	3.3%

Table 6: Index of dispersion for five cities

City	Austin	Singapore	Stockholm	San Francisco	Dallas
VMR	0.46	0.33	0.17	0.37	0.34

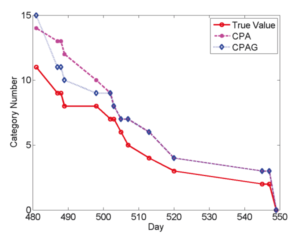
As compared to Austin, we find the degree of similarity between users is lowest in Singapore. In order to better describe this phenomenon, we employ the index of dispersion [7], or VMR (variance to mean ratio)[5], which is defined as

$$VMR = \frac{\sigma^2}{\mu}$$

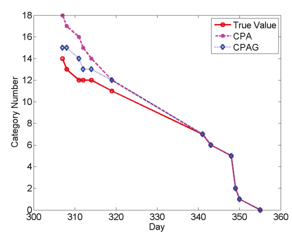
where μ is the mean and σ is the standard deviation of the similarity data. We calculate VMR values for the five cities, see Table 6. Compared to other four cities, the VMR value of Singapore is relatively low which means that it would be more of a hindrance than a help to cluster users in Singapore to obtain more data points for Chebyshev polynomial approximation.

Finally, we evaluate the performance of forecasting category numbers for every user in those three cities that achieved good results. For each city, we list three users' performance in Fig. 6, indicating that the category numbers of POIs that predicted via Algorithm 3 are closer to true values compared to Algorithm 2. Hence, grouping users in Algorithm 3 is indeed improving the Chebyshev approximation.

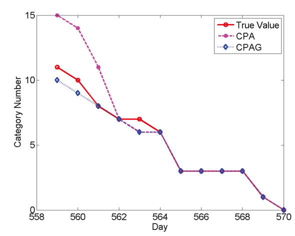
[5] It is a measure used to quantify whether a set of observed occurrences are clustered or dispersed compared to a standard statistical model.



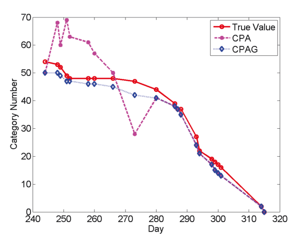
(a) User1-Stockholm



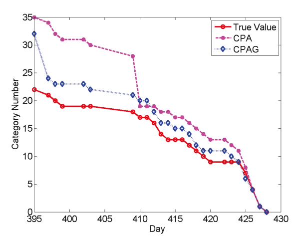
(b) User2-Stockholm



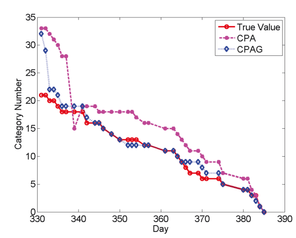
(c) User3-Stockholm



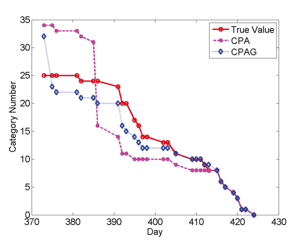
(d) User1-San Francisco



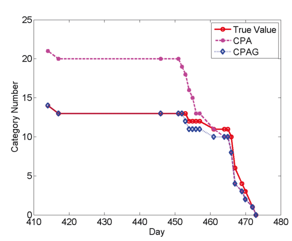
(e) User2-San Francisco



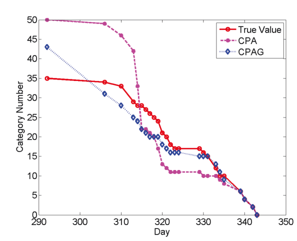
(f) User3-San Francisco



(g) User1-Dallas



(h) User2-Dallas



(i) User3-Dallas

Fig. 6: Performance of forecasting category numbers for three users in three cities

Effect of Varying Number of Chebyshev Polynomials This experiment is to study the effect of number of Chebyshev polynomials in our context. We report the MAE value for different numbers of Chebyshev polynomials on the dataset of Dallas via Algorithms 2 and 3 in Fig. 7. It shows that the MAE results maintain a relatively stable value no matter how large is the number of Chebyshev polynomials partially because the number of data points involved in Chebyshev approximation is still relatively smaller compared to [15], which is caused by the data sparsity.

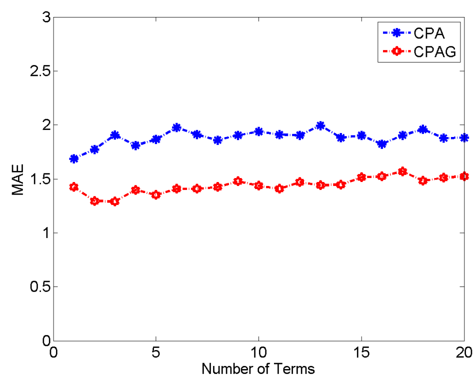


Fig. 7: Effect of Varying Number of Chebyshev Polynomials

6 Related Works

Collaborative filtering (CF) techniques are widely used for recommender system, see, e.g., [1,16,18]. The most common one is user-based system [12], which is based on the similarity among all users' check-in activities. Cosine similarity and Pearson correlation are usually used to measure similarity here. Ye *et al.* [16] studied different impacts under the framework of user-based CF on POI recommendation and improved the recommendation accuracy. They also discovered that users' preference plays a more important role than geographical and social influence. Different from [16], Liu *et al.* [13] proposed a framework that allows to capture the geographical influences on a user's check-in behaviors, which can effectively model the user mobility behavior, and dealt with the skewed distribution of check-in count data. The techniques are further improved by incorporating the temporal influence [10,17]. Yuan *et al.* [17] demonstrated that the check-in behavior at one time is more similar to some time slots than others, and then proposed a unified framework combining the temporal influence and spatial influence. Hence, the accuracy of the recommendation system is highly improved. Moreover, there are some other works to improve the recommendation quality.

Chen *et al.* [4] is the first work to consider information coverage of recommended POIs and the measurement diversity is introduced subsequently. More importantly, the parameter between diversity and accuracy of a recommended collection should be studied. This paper presents a way for choosing the key parameter to balance the relevance and information coverage. Accompany with the work [4], our work may contribute into attractive research on diversifying POI recommendations for each user.

7 Conclusion and Future Work

In this paper, we study users' check-in data on visiting location categories, which fits the law of diminishing marginal utility. We aim to balance the POI relevance and information coverage in the location category based POI recommendation. To tackle this problem, we establish a function between the balance parameter and the category number, and propose a method to approximate the function via Chebyshev polynomial approximation method. Then, the parameter value can be automatically adjusted according to each user's current preference. We conduct experiments over five real-world LBSN datasets. Results from extensive experiments demonstrate the expected performance of the proposed method. The performance also depends on the properties of the collected data in the recommendation. As for the next step, we plan to use the transfer learning techniques to improve the recommendation, e.g. the fitting β learnt from one city can be used to predict the preference of users in another city.

Acknowledgments. This work was partially supported by National Science Foundation of China (Grant No.: 11301436).

References

1. Adomavicius, G., Tuzhilin, A.: Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. IEEE TKDE, 17(6):734 – 749 (2005)
2. Bach, F.R., Jordan, M.I.: Learning Spectral Clustering. Advances in neural information processing systems, 16(2):2006, (2004)
3. Berjani, B., Strufe, T.: A recommendation system for spots in location-based online social networks. In Proceedings of the 4th Workshop on Social Network Systems, ACM, 1 – 6 (2011)
4. Chen, X., Zeng, Y., Gao, C., Qin, S., Xiang, Y., Dai, Y.: On Information Coverage for Location Category Based Point-of-Interest Recommendation. AAAI 2015, 37 – 43 (2015)
5. Cheng, C., Yang, H., King, I., Lyu, M.: Fused matrix factorization with geographical and social influence in location-based social networks. AAAI, Toronto, Canada (2012)
6. Cho, E., Myers, S.A., Leskovec, J.: Friendship and mobility: user movement in location-based social networks. In Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, pp. 1082 – 1090 (2011)

7. Cox, D. R., Lewis, P. A. W.: The Statistical Analysis of Series of Events. London: Methuen. (1966)
8. Cover, T.M., Thomas, J.A.: Elements of Information Theory. New York, USA, Wiley-Interscience (1991)
9. Curtis, P.O.W., Gerald, F.: Applied Numerical Analysis. Addison-Wesley Publishing Company, Menlo Park, CA (1992)
10. Gao, H., Tang, J., Hu, X., Liu, H.: Exploring Temporal Effects for Location Recommendation on Location-Based Social Networks. ACM Conference on Recommender Systems, 93 – 100 (2013)
11. Gil, A., Segura, J., Temme, N.M.: Numerical methods for special functions. Siam. (2007)
12. Herlocker, J.L., Konstan, J.A., Borchers, A., Riedl, J.: An algorithmic framework for performing collaborative filtering. In SIGIR, pp. 230 – 237 (1999)
13. Liu, B., Fu, Y., Yao, Z., Xiong, H.: Learning Geographical Preferences for Point-of-Interest Recommendation. Acm Sigkdd International Conference on Knowledge Discovery and Data Mining, pp. 1043 – 1051 (2013)
14. Mankiw, N.G.: Principles of Microeconomics (seventh Edition). Cengage Learning, Stamford, USA (2013)
15. Saha, S., Biswas, A., Sen, S.: Modeling Opponent Decision in Repeated One-shot Negotiations. Autonomous Agents and Multiagent Systems/International Conference on Autonomous Agents - AAMAS(Agents), pp. 397 – 403 (2005)
16. Ye, M., Yin, P., Lee, W.-C., Lee, D.-L.: Exploiting Geographical Influence for Collaborative Point-of-Interest Recommendation. International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 325 – 334 (2011)
17. Yuan, Q., Cong, G., Ma, Z., Sun, A., Thalmann, N.M.: Time-aware Point-of-interest Recommendation. In Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 363 – 372 (2013)
18. Zheng, V.W., Zheng, Y., Xie, X., Yang, Q.: Collaborative location and activity recommendations with gps history data. In Proceedings of the 19th International Conference on World Wide Web(WWW), 1029 – 1038 (2010)